Graph-based Event Extraction from Twitter

Amosse Edouard, Elena Cabrio, Sara Tonelli, Nhan Le Thanh

Graph-based Event Extraction from Twitter

Amosse Edouard
Inria, CNRS, I3S
Nice, France
edouard@unice.fr

Elena Cabrio
Inria, CNRS, I3S
Nice, France
cabrio@unice.fr

Sara Tonelli
Fondazione Bruno Kessler
Trento, Italy
satonelli@fbk.eu

Nhan Le-Than
Inria, CNRS, I3S
Nice, France
le-than@unice.fr

Abstract

Event detection on Twitter has become an attractive and challenging research field due to the popularity and the peculiarities of tweets. Detecting which tweets describe a specific event and clustering them is one of the main challenging tasks related to Social Media currently addressed in the NLP community. Existing approaches have mainly focused on detecting spikes in clusters around specific keywords or Named Entities (NE). However, one of the main drawbacks of such approaches is the difficulty in understanding when the same keywords describe different events. In this paper, we propose a novel approach that exploits NE mentions in tweets and their entity context to create a temporal event graph. Then, using simple graph theory techniques and a PageRank-like algorithm, we process the event graphs to detect clusters of tweets describing the same events. Experiments on two gold standard datasets show that our approach achieves state-of-the-art results both in terms of evaluation performances and the quality of the detected events.

1 Introduction

Twitter has become a valuable source of timely information covering topics from every corner of the world. For this reason, NLP researchers have shown growing interest in mining knowledge from Twitter data. As a result, several approaches have been proposed to build applications over tweets, e.g. to extract structured representations/summary of newsworthy events (McMinn and Jose, 2015; Katragadda et al., 2017), or to carry out sentiment analysis on the short messages to study users reactions to certain topics (Agarwal et al., 2011; Kouloumpis et al., 2011). However, processing tweets is a non trivial task, given their peculiarities of being at most 140 characters long, containing little contextual information, misspelled words and jargon. Moreover, information in Twitter stream is continuously changing because of the real-time nature of social media, while at the same time there might be a high volume of redundant messages referring to the same issue or event.

In this work, we focus on event extraction from Twitter, consisting in the automated clustering of tweets related to the same event based on relevant information such as time and participants. Our approach is in line with the event definition provided by (Dou et al., 2012), i.e. "*an occurrence causing change in the volume of text data that discusses the associated topic at a specific time. This occurrence is characterized by topic and time, and often associated with entities such as people and location*".

Existing approaches to the task create clusters of tweets around event-related keywords (Parikh and Karlapalem, 2013), or NEs (McMinn and Jose, 2015). However, such approaches fail *i)* to capture events that do not generate spikes in the volume of tweets, for instance "Richard Bowes, victim of London riots, dies in hospital"; and *ii)* to distinguish between events that involve the same NEs and keywords, as for instance "the shoot of Malala Yousafzai, the 14-year old Pakistani activist" and "her successful surgery" later on. Other approaches model the relationships between terms contained in the tweets relying on a graph representation (Katragadda et al., 2016), and retain the nodes with the highest number of edges as event candidates. However, the main drawbacks of these approaches are that *i)* they generate highly dense graphs, and *ii)* trending terms not related to events

may be considered as event candidates.

To address such limitations, in this work we propose an unsupervised approach to detect open-domain events on Twitter, where the stream of tweets is represented through temporal event graphs, modeling the relations between NEs and the terms that surround their mentions in the tweets.

The paper is organized as follows: Section 2 presents related work and the state of the art approaches. Then, Section 3 describes the approach we propose for event detection on Twitter. Section 4 reports on the experiments we carried out to evaluate our work, and to compare it with existing approaches. Section 5 concludes the paper and presents direction for future works.

2 Related Work

Existing approaches to extract events from tweets can be divided into two main categories, namely closed-domain and open-domain event detection systems (Atefeh and Khreich, 2015). In the closed-domain, approaches are mainly focused on extracting a particular type of event, as for instance natural disasters (Panem et al., 2014). Works in the closed-domain scenario are usually cast as supervised classification tasks that rely on keywords to extract event-related messages from Twitter (Wang et al., 2012), to recognize event patterns (Popescu et al., 2011) or to define labels for training a classifier (Anantharam et al., 2015; Sakaki et al., 2010).

The open-domain scenario is more challenging, since it is not limited to a specific type of event and usually relies on unsupervised models. Among the works applying an unsupervised approach to event detection on Twitter, (McMinn and Jose, 2015) create event clusters from tweets using NE mentions as central terms driving the clusters. Thus, tweets mentioning the same entities are grouped together in a single cluster. Experiments on a public dataset of tweets show that this strategy outperforms other approaches such as Latent Sensitive Hashing (Petrović et al., 2010). Similarly, (Hasan et al., 2016) create clusters based on cosine similarity among tweets. Both works do not consider the temporal aspect of events and fail to capture terms or entities involved in different events at different time periods, as the aforementioned example on the events related to the shoot of *Malala Yousafzai* and her surgery.

(Katragadda et al., 2016, 2017) use graphs to model relationships between terms in tweets at different time windows. First, a graph is created based on the set of links between terms in tweets, where terms are considered as connected according to the order of their appearance in the text independently of their syntactic or semantic relations. Then, the graph is pruned to remove terms that are less frequent than a given threshold. Finally, clusters in the graph are evaluated in order to determine whether or not they are credible, where the credibility of an event is determined by their presence or not in other time windows. Differently from them, we create event graphs from terms that appear in the NE context, which contributes in reducing the density of the event graphs by considering event-related features.

Very recently, (Zhou et al., 2017) use a non-parametric Bayesian Mixture Model leveraged with word embeddings to create event clusters from tweets. In this approach, events are modeled as a 4-tuple $\langle y, l, k, d \rangle$ modeling non-location NEs, location NEs, event keywords and date. Each component of the quadruple is generated from a multinomial distribution computed with Dirichlet process. The work was focused on detecting events given a set of event-related tweets, which is however not applicable to a real scenario, where the stream of tweets can also contain messages that are not event-related. This scenario is simulated in the second experiment presented in this paper.

3 Approach description

In this section, we describe our approach for detecting open-domain events on tweets. The proposed approach is based on graph theory to model relations between terms in tweets. The pipeline consists of the following components: Tweet preprocessing, Named Entity recognition and linking, graph creation, graph partitioning, event detection and event merging. Each step is described in the following subsections.

3.1 Tweet Preprocessing

The workflow starts by collecting tweets published during a fixed time window, which can be set as input parameter (e.g. 1 hour). Then, we apply common text preprocessing routines to clean the input tweets. We use TweetMotifs (O'Connor et al., 2010), a specific tokenizer for tweets, which treats

hashtags, user mentions and emoticons as single tokens. Then, we remove the retweets, URLs, non ASCII characters and emoticons. It is worth mentioning that at this stage we do not perform stop word removal since stop words can be part of NEs (e.g. United States of America). As for hashtags, we define a set of hand-crafted rules to segment them into meaningful terms. As an example, we use the capital character to break hashtags in terms (e.g. *#presidentialDebate is replaced by 'presidential debate'*).

Since Tweets often contain misspelled terms, we try to correct them using the Symmetric Delete Spelling Correction algorithm (SymSpell)[1], which matches misspelled tokens with Wordnet synsets (Fellbaum, 1998).

3.2 Named Entity Recognition and Linking

We use NERD-ML (Van Erp et al., 2013), a Twitter specific Named Entity Recognizer (NER) tool, to extract NE mentions in the tweets. Our choice is motivated by (Derczynski et al., 2015), showing that NERD-ML is among the best performing tools for NER on Twitter data. Besides, NERD-ML not only recognizes the most common entity types (i.e. Person, Organization and Location), but tries also to link any term listed in external knowledge bases such as DBpedia[2] or Wikipedia. These are then associated with semantic classes in the NERD ontology. We report in Table 1 an example tweet after the NE recognition and linking steps.

The EU has won a noble peace prize! I'm guessing merkel will go and accept it
The European_Union has won a Noble_Peace_Prize! I'm guessing Angela_Merkel will go and accept it

Table 1: Output of the Entity Recognition and Linking module on tweets: entity mentions are normalized after linking.

3.3 Graph Generation

Previous works using graph-based methods to model relations between terms in text considered all terms in the input document as nodes and used their position in text to set edges (Andersen et al., 2006; Xu et al., 2013). Such approaches may generate a dense graph, which generally requires high computational costs to be processed.

In this work, we assume that the terms surrounding the mention of a NE in a tweet define its context (Nugroho et al., 2015). Thus, we rely on the NE context to create the event graphs, built as follows:

- Nodes : We consider NE and k terms that precede and follow their mention in a tweet as nodes, where $k > 1$ is the number of terms surrounding a NE to consider while building the NE context.

- Edges : Nodes in the graph are connected by an edge if they co-occur in the context of a NE.

- Weight: The weight of the edges is the number of co-occurrences between terms in the NE context. In addition, each edge maintains as a property the list of tweets from which the relationship is observed.

Formally, let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be a directed graph (or digraph) with a set of vertices $\mathcal{V}$ and edges $\mathcal{E}$, such that $\mathcal{E} \subset \mathcal{V}x\mathcal{V}$. For any $\mathcal{V}_i \in \mathcal{V}$, let $In(\mathcal{V}_i)$ be the set of vertices that point to $\mathcal{V}_i$ (i.e. predecessors), and $Out(\mathcal{V}_i)$ be the set of vertices that $\mathcal{V}_i$ points to (i.e. successors).

Let $\mathcal{E}_i = (\mathcal{V}_j, \mathcal{V}_k)$ be an edge that connects node $\mathcal{V}_j$ to $\mathcal{V}_k$, we define ω_{ij} as the weight of $\mathcal{E}_i$, which is represented by the number of times relationships between $\mathcal{V}_j$ and $\mathcal{V}_k$ is observed in tweets published during a time window. An example of the graph created on 2011-07-07 with tweets related to the famine in Somalia and space shuttle to Mars is shown in Figure 1.

3.4 Graph Partitioning

At this stage, an event graph is generated to model relationships between terms in the NE contexts. We apply graph theory to partition the graph into sub-graphs, which will be considered as event candidates. Tweets related to the same events usually share a few common keywords, while tweets that are not related to events or those related to different events are usually characterized by different keywords (McMinn and Jose, 2015). In the event graphs, this phenomenon is expressed by stronger links between nodes related to the same event. In other words, the weight of edges that connect terms from tweets related to similar events

[1] https://github.com/wolfgarbe/symspell
[2] http://dbpedia.com/

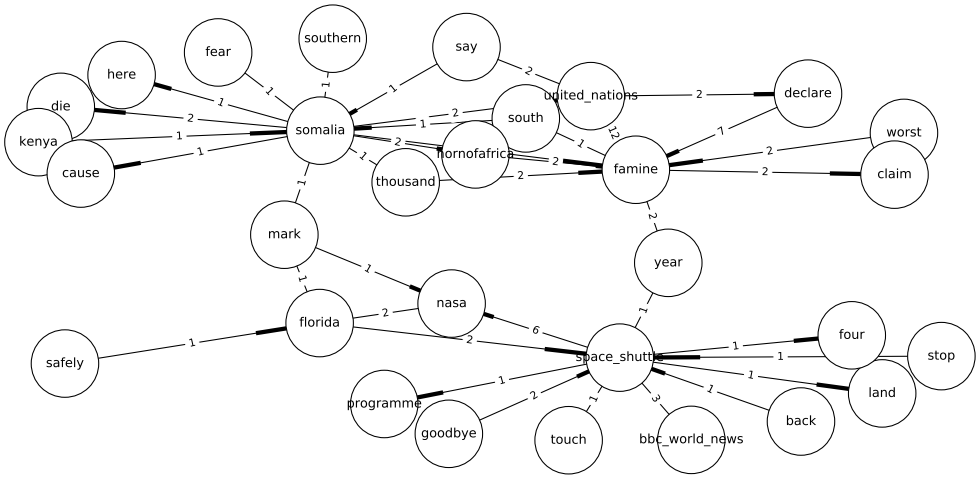


Figure 1: Graph generated on day "2011-07-07" from a sample of tweets related to the events about the famine in Somalia and the space shuttle to Mars.

are higher than edges between nodes that connect terms from tweets related to different events. The graph partitioning purpose is to identify such edges that, if removed, will split the large graph $\mathcal{G}$ into sub-graphs.

Let $\mathcal{E} = \{(\mathcal{V}_1, \mathcal{W}_1), (\mathcal{V}_2, \mathcal{W}_2), ..., (\mathcal{V}_n, \mathcal{W}_n)\}$ be a set of pair of vertices in a strongly connected graph $\mathcal{G}$. We define λ as the least number of edges whose deletion from $\mathcal{G}$ would split $\mathcal{G}$ into connected sub-graphs. Similarly, we define the edge-connectivity $\lambda(\mathcal{G})$ of $\mathcal{G}$ of an edge set $\mathcal{S} \subset \mathcal{E}$ as the least cardinality $|\mathcal{S}|$ such that $\mathcal{G} - \mathcal{S}$ is no longer strongly connected. For instance, given the graph in Figure 1 as input, the deletion of edges "mark/somalia" and "year/famine" will create two strongly connected sub-graphs, where the first one contains keywords related to "famine in Somalia" and other contains keywords related to "The space shuttle to Mars".

3.5 Event Detection

In our event detection approach, we assume that events from different sub-graphs are not related to each other. Thus, in the event detection sub-module, each sub-graph is processed separately. In a study on local partitioning, Andersen et al. (2006) show that a good partition of a graph can be obtained by separating high-ranked vertices from low-ranked ones, if the nodes in the graph have distinguishable values. Similar to Mihalcea and Tarau (2004), we use a PageRank-like algorithm (Brin and Page, 1998) to rank vertices in the event-graph as follows :

$$S(V_i) = ((1-d) + d \sum_{v_j \in In(V_i)} \frac{w_{ji}}{\sum_{v_k \in Out(V_k)} \omega_{jk}} S(V_j))\epsilon_i \quad (1)$$

where ω_{ij} is the weight of edge connecting V_i to V_j, d a dumping factor usually set to 0.85 (Brin and Page, 1998) and ϵ_i a penalization parameter for node i. In previous approaches (Mihalcea and Tarau, 2004), the penalization parameter is considered as a uniform distribution; instead, we define the penalization parameter of a node according to its tf-idf score. Due to redundant information in tweets, the score of the nodes can be biased by the trending terms in different time windows. Thus, we use the tf-idf score to reduce the impact of trending terms in the collection of tweets. Before computing the score with equation 1, we assign an initial value $\tau = 1/n$ to each vertex in the graph, where n is the total number of nodes in the graph. Then, for each node, the computation iterates until the desired degree of convergence is reached. The degree of convergence of a node can be obtained by computing the difference between the score at the current iteration and at the previous iteration, which we set to 0.0001 (Brin and Page, 1998). Notice that the final salience score of each node is not affected by the choice of the initial value assigned to each node in the graph, but rather by the weight

of the edges (Mihalcea and Tarau, 2004).

As shown in Algorithm 1, we start by splitting the vertex set into high-ranked and low-ranked vertices based on a gauged parameter α (Line 3). Next, we process the vertices in the high-ranked subset starting from the highest ones, and for each candidate we select the highest weighted predecessors and successors as keywords for event candidates (Lines 4-9). After removing the edges between the keywords from the graph, if it becomes disconnected, we also consider the disconnected nodes as keywords for the event candidate (Lines 10-13). Based on the semantic class provided by the NER tool (see Section 3.2), we divide the keywords related to an event in the following subsets: *what* (i.e., the type of the event), *where* (i.e., the location in which the event happens), *who* (i.e., the person or organization involved). As for the date, we select the oldest tweets that report the event. We recall that each edge contains a list of tweets from which the relationship is obtained. Thus, for each connected node in the keywords of an event, we consider the tweets as related to the event candidate (Lines 17-19).

In the second stage of Algorithm 1, we further preprocess the event candidates to remove noise and duplicate events. First, we merge duplicate event candidates (Lines 22-35). Event candidates are considered as duplicate if they share common terms and have the same location or participants in the considered time window. When two event candidates are found as duplicate, they are merged into a new event built from the combination of terms and entities of the two event candidates. An event is considered as valid if at least a NE is involved, and if it occurs in a minimum number of tweets provided as input parameter.

3.6 Event Merging

It is common to observe in the Twitter stream mentions of the same event in different time slices (e.g. hours, days, ...). Thus, we found it important to detect and merge duplicated events. We consider events in different time-windows as duplicate if they contain the same keywords, entities (e.g. person, organization, location) in an interval of k days, where k is an input parameter. When a new event is found as duplicate, we merge it with the previous detected event.

Algorithm 1 Algorithm to process a given event-graph to retrieve important sub-events.

1: **function** GRAPH_PROCESSING(G, α)
2: $E = \emptyset$
3: $H = \{v_i \in vertex(G) if score(v_i) >= \alpha\}$ ▷ Equation 1
4: **while** $H \neq \emptyset$ **do**
5: $G' = G.copy()$
6: $v_i = H.pop()$
7: $p = max(W_j \in In(v_i)$
8: $s = max(W_j \in Out(v_i)$
9: $keywords = set(p, v_i, s)$
10: $G'.remove_edges((p, v_i), (v_i, s))$
11: **if** $notG'.connected()$ **then**
12: $append(keywords, disc\_vertices(G'))$
13: **end if**
14: $who = person||organization \in keywords$
15: $where = location \in keywords$
16: $what = keywords - who - where$
17: $tweets = tweet\_from(keywords)$
18: $when = oldest(tweets, date)$
19: $event =< what, who, where, when >$
20: $append(E, event)$
21: **end while**
22: **for** $e \in E$ **do**
23: **for** $e' in E$ **do**
24: **if** $what(e) \cap what(e')$ **then**
25: **if** $who(e) \cap who(e')$ **then**
26: $merge(e, e')$
27: **end if**
28: **if** $where(e) \cap where(e')$ **then**
29: $merge(e, e')$
30: **end if**
31: **end if**
32: **end for**
33: **if** $notwho(e) or notwhere(e)$ **then**
34: $discard(E, e)$
35: **end if**
36: **end for**
return E
37: **end function**

4 Experiments

In this section, we describe the experiments carried out to validate our approach. Given a set of tweets, the goal of these experiments is to cluster such tweets so that each cluster corresponds to a fine-grained event such as "Death of Amy Winehouse" or "Presidential debate between Obama and Romney during the US presidential election". We first describe the datasets, then we present the experimental setting. This section ends with a comparison of the obtained experimental results with state-of-the-art approaches.

4.1 Dataset

We test our approach on two gold standard corpora: the First Story Detection (FSD) corpus (Petrović et al., 2012) and the EVENT2012 corpus (McMinn et al., 2013).

FSD The corpus was collected from the Twitter streaming API[3] between 7th July and 12th September 2011. Human annotators annotated 3,035 tweets as related to 27 major events occurred in that period. The corpus covers various events such as the death of Amy Winehouse, the earthquake in Virginia, or the crash of the Russian hockey team plane. After removing tweets that are no more available, we are left with 2,342 tweets related to one out of the 27 events. To reproduce the same dataset used by other state-of-the-art approaches, we consider only those events mentioned in more than 15 tweets. Thus, the final dataset contains 2,295 tweets describing 20 events.

EVENT2012 A corpus of 120 million tweets collected from October to November 2012 from the Twitter streaming API, of which 159,952 tweets were labeled as event-related. 506 event types were gathered from the Wikipedia Current Event Portal, and Amazon Mechanical Turk was used to annotate each tweet with one of such event types. Events covered by this dataset include, for example, the US presidential election results, or the Chemistry Nobel prize. After removing tweets that are no longer available, our final dataset contains ∼43 million tweets from which 152,758 are related to events.

4.2 Experimental Setting

For each dataset, we compare our approach with state-of-the-art approaches. For the FSD dataset, we compare with LEM Bayesian model (Zhou et al., 2011) and DPEMM Bayesian model enriched with word embeddings (Zhou et al., 2017). For the EVENT2012 dataset, we compare our results with Named Entity-Based Event Detection approach (NEED) (McMinn and Jose, 2015) and Event Detection Onset (EDO) (Katragadda et al., 2016).

In order to simulate a real scenario where tweets are continuously added to a stream, we simulate the Twitter stream with a client-server architecture which pushes tweets according to their creation date. We evaluate our approach in two different scenarios: in the first scenario, we consider tweets from the FSD dataset that are related to events and we classify them into fine-grained event clusters. In the second scenario, we adopt a more realistic approach in that we consider all the tweets from the EVENT2012 dataset (i.e event-related and not event-related ones), and we classify them into event clusters, discarding those that are not related to events.

Our approach requires a few parameters to be provided as input. In the experiments reported in this paper, we process the input stream with fixed time-window $w = 1$ hour. The minimum number of tweets for event candidates is set to $n = 5$. Finally, we empirically choose $t = 3$ days as the interval of validity for the detected events.

4.3 Results

Performance is evaluated both in terms of P/R/F1 and the quality of the detected events, i.e. cluster purity.

Precision is computed as the ratio between the number of events (i.e. tweet clusters) correctly classified and the number of detected events. Recall is calculated by taking the ratio between the number of events correctly classified and the number of events in the ground truth.

4.3.1 Results on the FSD dataset

In this scenario, we consider an event as correctly classified if *all* the tweets in that cluster belong to the same event in the gold standard, otherwise the event is considered as misclassified. In addition, due to the low number of tweets, we set the gauged parameter $\alpha = 0.5$ as the minimum score for nodes in the graph to be considered as useful for events. Table 2 shows the experimental results yielded by our approach in comparison to state-of-the-art approaches. Our approach outperforms the others, improving the F-score by 0.07 points w.r.t. DPEMM and by 0.13 w.r.t. LEM. Example of events detected by our approach are for instance "space shuttle on Mars by NASA on 2011-07-21", for which we detect highly informative terms such as "space shuttle", the correct place "Kennedy Space Center" as well as the participants involved such as "NASA".

Approach	Precision	Recall	F-measure
LEM	0.792	0.850	0.820
DPEMM	0.862	0.900	0.880
Our Approach	0.950	0.950	0.950

Table 2: Evaluation results on the FSD dataset.

Furthermore, we evaluate the quality of the events, i.e. the clusters, in terms of purity, where the purity of an event is based on the number of

[3] dev.twitter.com/streaming/overview

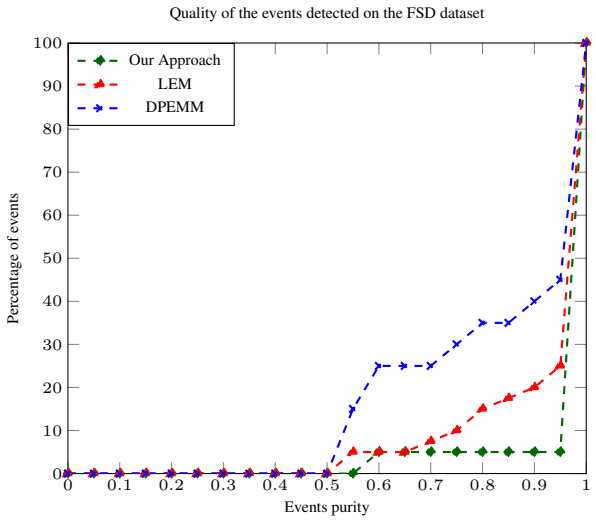


Figure 2: Purity of the events detected by our approach, LEM and DPEMM on the FSD dataset. The y-axis denotes the percentage of events and the x-axis the purity of the events.

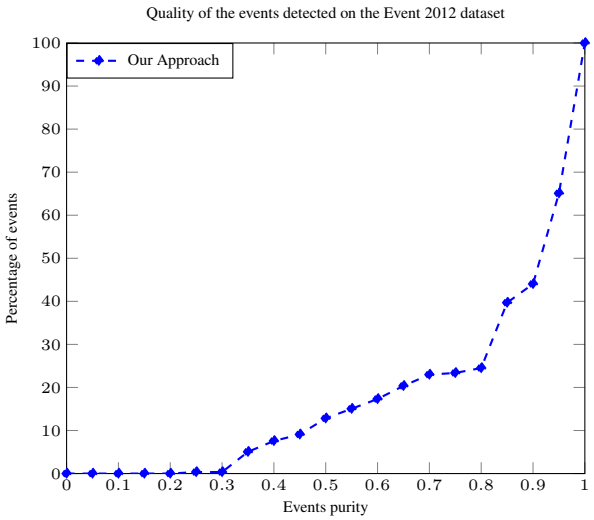


Figure 3: Purity of the events detected by our approach on the event 2012 dataset. The y-axis denotes the percentage of events and the x-axis the purity of the events.

tweets correctly classified in the cluster and the number of misclassified tweets. More specifically, purity is computed as: $P_e = \frac{n_e}{n}$, where n_e is the number of tweets correctly classified and n the total number of tweets classified in that cluster. Figure 2 reports the purity of our approach compared to LEM and DPEMM, where each point (x, y) denotes the percentage of events having purity less than x. It can be observed that 5% of the events detected as well as DPEMM have purity less than 0.65 compared to 25% for LEM, while 95% of the events detected have purity higher than 0.95 compared to 75% for DPEMM and 55% for LEM.

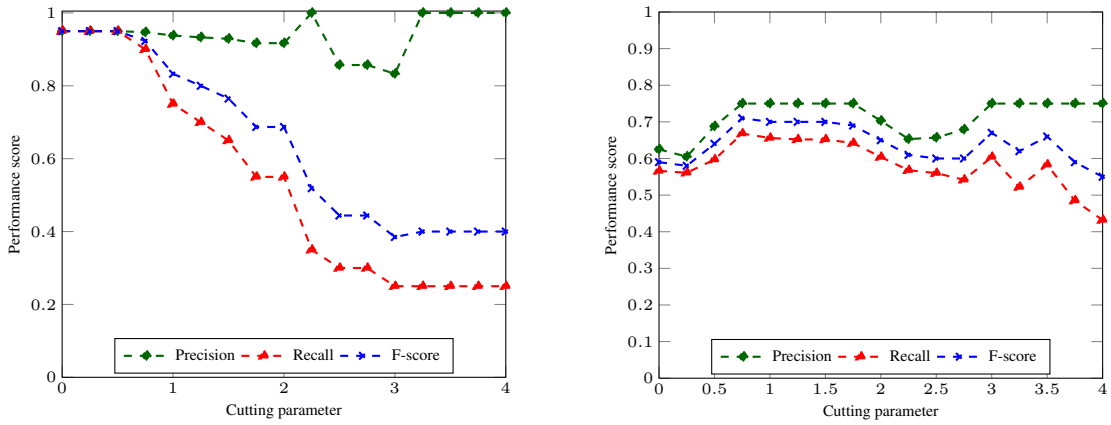
4.3.2 Results on the EVENT2012 dataset

We also evaluate our approach on the EVENT2012 dataset using a more realistic scenario in which all the tweets (i.e. events related and non-event related tweets) are considered. Compared to the FSD dataset, the EVENT2012 dataset has more events and tweets and thus a larger vocabulary. We set the cutting parameter $\alpha = 0.75$ as the minimum score of nodes in the graph to be considered as important for events. We further detail the importance of the parameters α in Section 4.4. Also, since we include both event-related and not event-related tweets, we consider an event as correct if 80% of the tweets belong to the same event in the ground truth. Table 3 reports on the experimental results compared to the NEED and EDO approaches. In general, our approach improves the f-score by 0.07 points w.r.t. EDO and 0.23 points w.r.t. NEED. After a manual check of the output, we noticed that some issues with precision may depend on the quality of the dataset, since some tweets related to events were not annotated as such in the gold standard. For example, we found that 9,010 tweets related to "BET hip hop award" were not annotated. The same was found for tweets concerning large events such as "the Presidential debate between Obama and Romney" or the "shooting of Malala Yousafzai, the 14-year old activist for human rights in Pakistan".

We also evaluate the purity of the events detected by our approach (Figure 3). We can observe that the quality of the detected events is lower than for the events detected on the FSD dataset. For instance, more than 20% of the detected events have purity *lower* than 0.7. As expected, event purity is mainly affected by the inclusion in the clusters of non event-related tweets.

Approach	Precision	Recall	F-measure
NEED	0.636	0.383	0.478
EDO	0.754	0.512	0.638
Our Approach	0,750	0.668	0.710

Table 3: Evaluation results on the EVENT2012 dataset.



(a) Effect on FSD

(b) Effect on EVENT2012

Figure 4: Effect of the cutting parameter (α) on the performance of our approach.

4.4 Effect of the Cutting Parameter

We further experiment on the impact of the dangling parameter on the output of our model. The dangling parameter α is used to separate the nodes of the event graph into high-ranked and low-ranked nodes, where the high-ranked nodes are used to extract keywords related to event candidates. We experiment different values for "α" and we evaluate their impact on the performance of our approach on both datasets.

In Figure 4a we show the performance of our model for $0 < \alpha \leq 4$ on the FSD dataset. We observe that higher value of α gives higher precision while lowering the recall. More specifically, for $\alpha \geq 3$ we obtain 100% precision and recall lower than 50%. On the other hand, the best performance is obtained for $\alpha \leq 0.5$. Since the FSD dataset contains $\sim 6,000$ unique words, at each time window the generated graph is strongly connected, thus the average minimum score of the nodes is higher than 0.5. For values higher than 0.5, important terms referring to events are ignored, mainly when they are related to events that do not generate a high volume of tweets. In our experiments, we also observe that higher values of α mostly affect the recognition of events with low number of tweets.

Figure 4b shows the performance of our model for different values of α on the EVENT2012 dataset. We observe that for different values of α, both precision and recall are affected. More specifically, the recall of the model tends to decrease for lower values of α. Without edge cutting (i.e. $\alpha = 0$), the recall of our model is similar to EDO. Overall, the impact of α is bigger on the EVENT2012 dataset than on FSD dataset. The variation of precision and recall curves is smaller for consecutive values of α w.r.t. to FSD. There are two main reasons for that: *i)* the EVENT2012 dataset has a richer vocabulary, and *ii)* many events in the EVENT2012 dataset are similar to each other.

5 Conclusions and Future Works

In this paper, we described a model for detecting open-domain events from tweets by modeling relationships between NE mentions and terms in a directed graph. The proposed approach is unsupervised and can automatically detect fine-grained events without prior knowledge of the number or type of events. Our experiments on two gold-standard datasets show that the approach yields state-of-the-art results. In the future, we plan to investigate whether linking terms to ontologies (e.g. DBpedia, YAGO) can help in detecting different mentions of the same entity, for instance "German chancellor" and "Angela Merkel". This can be used to reduce the density of the event graph. Another possible improvement would be to enrich the content of the tweets with information from external web pages resolving the URLs in the tweets.

References

Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. 2011. Sentiment analysis of twitter data. In *Proceedings of the workshop on languages in social media*. Association for Computational Linguistics, pages 30–38.

Pramod Anantharam, Payam Barnaghi, Krishnaprasad Thirunarayan, and Amit Sheth. 2015. Extract-

ing city traffic events from social streams. *ACM Transactions on Intelligent Systems and Technology (TIST)* 6(4):43.

Reid Andersen, Fan Chung, and Kevin Lang. 2006. Local graph partitioning using pagerank vectors. In *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*. IEEE, pages 475–486.

Farzindar Atefeh and Wael Khreich. 2015. A survey of techniques for event detection in twitter. *Computational Intelligence* 31(1):132–164.

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems* 30(1):107–117.

Leon Derczynski, Diana Maynard, Giuseppe Rizzo, Marieke van Erp, Genevieve Gorrell, Raphaël Troncy, Johann Petrak, and Kalina Bontcheva. 2015. Analysis of named entity recognition and linking for tweets. *Information Processing & Management* 51(2):32–49.

Wenwen Dou, K Wang, William Ribarsky, and Michelle Zhou. 2012. Event detection in social media data. In *IEEE VisWeek Workshop on Interactive Visual Text Analytics-Task Driven Analytics of Social Media Content*. pages 971–980.

C. Fellbaum. 1998. *WordNet. An Electronic Lexical Database*. MIT Press.

Mahmud Hasan, Mehmet A Orgun, and Rolf Schwitter. 2016. Twitternews: real time event detection from the twitter data stream. *PeerJ PrePrints* 4:e2297v1.

Satya Katragadda, Ryan Benton, and Vijay Raghavan. 2017. Framework for real-time event detection using multiple social media sources. In *Proceedings of the 50th Hawaii International Conference on System Sciences*.

Satya Katragadda, Shahid Virani, Ryan Benton, and Vijay Raghavan. 2016. Detection of event onset using twitter. In *Neural Networks (IJCNN), 2016 International Joint Conference on*. IEEE, pages 1539–1546.

Efthymios Kouloumpis, Theresa Wilson, and Johanna D Moore. 2011. Twitter sentiment analysis: The good the bad and the omg! *Icwsm* 11(538-541):164.

Andrew J McMinn and Joemon M Jose. 2015. Real-time entity-based event detection for twitter. In *International Conference of the Cross-Language Evaluation Forum for European Languages*. Springer, pages 65–77.

Andrew J. McMinn, Yashar Moshfeghi, and Joemon M Jose. 2013. Building a large-scale corpus for evaluating event detection on twitter. In *Proceedings of CIKM*. ACM, pages 409–418.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into texts. Association for Computational Linguistics.

Robertus Nugroho, Weiliang Zhao, Jian Yang, Cecile Paris, Surya Nepal, and Yan Mei. 2015. Time-sensitive topic derivation in twitter. In *International Conference on Web Information Systems Engineering*. Springer, pages 138–152.

Brendan O'Connor, Michel Krieger, and David Ahn. 2010. Tweetmotif: Exploratory search and topic summarization for twitter. In *ICWSM*. pages 384–385.

Sandeep Panem, Manish Gupta, and Vasudeva Varma. 2014. Structured information extraction from natural disaster events on twitter. In *Proceedings of the 5th International Workshop on Web-scale Knowledge Representation Retrieval & Reasoning*. ACM, pages 1–8.

Ruchi Parikh and Kamalakar Karlapalem. 2013. Et: events from tweets. In *Proceedings of the 22nd International Conference on World Wide Web*. ACM, pages 613–620.

Saša Petrović, Miles Osborne, and Victor Lavrenko. 2010. Streaming first story detection with application to twitter. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 181–189.

Saša Petrović, Miles Osborne, and Victor Lavrenko. 2012. Using paraphrases for improving first story detection in news and twitter. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. pages 338–346.

Ana-Maria Popescu, Marco Pennacchiotti, and Deepa Paranjpe. 2011. Extracting events and event descriptions from twitter. In *Proceedings of the 20th international conference companion on World wide web*. ACM, pages 105–106.

Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. 2010. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*. ACM, pages 851–860.

Marieke Van Erp, Giuseppe Rizzo, and Raphaël Troncy. 2013. Learning with the web: Spotting named entities on the intersection of nerd and machine learning. In *# MSM*. Citeseer, pages 27–30.

Xiaofeng Wang, Matthew S Gerber, and Donald E Brown. 2012. Automatic crime prediction using events extracted from twitter posts. In *Social Computing, Behavioral-Cultural Modeling and Prediction*, Springer, pages 231–238.

Wei Xu, Ralph Grishman, Adam Meyers, and Alan Ritter. 2013. A preliminary study of tweet summarization using information extraction. *NAACL 2013* page 20.

Deyu Zhou, Liangyu Chen, and Yulan He. 2011. A simple bayesian modelling approach to event extraction from twitter. *Atlantis* page 0.

Deyu Zhou, Xuan Zhang, and Yulan He. 2017. Event extraction from Twitter using Non-Parametric Bayesian Mixture Model with Word Embeddings. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*. Valencia, Spain, pages 808–817.